# Le recrutement IT en Algérie

## Retour d'expérience d'un développeur

Yassine Chaouche – yacinechaouche@yahoo.com -- http://ychaouche.wikispot.org

# RECRUTEZ

Faites-le au moins une fois dans votre vie, même si ce n'est pas vous qui êtes chargé de le faire. Recrutez. Vous verrez : c'est amusant de passer de l'autre côté du bureau. Vous allez me dire : mais nous n'avons pas le bon profil pour faire ce job, nous n'en avons pas les compétences. Ce n'est pas grave, demandez quand même à votre hiérarchie d'être impliqué dans le processus de recrutement. Vous n'aurez peut-être pas tout le pouvoir de décision sur les candidats, mais vous pourrez au moins aider dans la sélection de ceux-ci selon les critères que vous maîtrisez le mieux. Expliquez-leur que vous connaissez très bien le poste proposé, les compétences requises, le type de profil correspondant. Dans mon cas, ça a marché. On m'a même dit que c'était obligatoire de participer au processus de recrutement. C'est peut être un peu parce qu'en cette période, nous n'avions pas de réel DRH au sein de notre entreprise, et nous n'avions pas jugé utile de faire appel à un cabinet de recrutement spécialisé. Ce n'était pas pour un poste de haute qualification, seulement un développeur web. J'étais moi même développeur web, je n'avais donc aucune difficulté pour recruter un pair.

Dans ce document, je vais essayer de vous raconter comment ça c'est passé. Si mon histoire ne vous a pas plu, je serai très intéressé par vos critiques, laissez-moi un email. Si elle vous a plu et que vous voulez me le faire savoir, ça sera tout aussi sympathique :)

Bonne lecture !

# Les 8 étapes de mon processus de recrutement

1. **Établir une grille d'évaluation**
2. **Chercher les candidats**
3. **Premier filtrage : lecture des C.V**
4. **Deuxième filtrage : première entretien téléphonique**
5. **Troisième filtrage : deuxième entretien de pré-embauche, test des aptitudes techniques**
6. **Tri des candidats par score et sélection du candidat**
7. **Troisième entretien avec la DRH**
8. **Quatrième entretien avec le D.G, signature du contrat**

## (1) Etablir une grille d'évaluation

Les compétences techniques ne suffisent pas à elles seules à discriminer un bon candidat. Il peut être bon techniquement mais avoir de graves problèmes de communication. Il peut être très intelligent mais avec une attitude arrogante, insolente. Il peut avoir plusieurs années d'expérience mais pas assez diversifiée.

Une des premières choses que vous devez faire *avant* de recruter est de savoir comment vous allez classer vos candidats : il vous faut ***une grille d'évaluation***. Pour cela, j'utilise une méthode simple : une simple moyenne de notes pondérées sur chaque critère de sélection. Voici comment je procède :

- Je commence par lister en vrac tout ce que je juge pertinent comme critère.
- Je les ordonne par ordre d'importance, il y a aussi une méthode pour ça que je détaillerai au moment où on parlera du tri.
- J'affecte des coefficients en fonction du rang, en commençant par 1 et en augmentant de 1 à chaque palier. On peut augmenter de plus de 1 si on estime que le critère suivant est beaucoup plus important que le précédent. Si deux critères sont jugés égaux, je leur donne la même note.
- J'évalue le candidat pour chaque compétence et lui donne une note sur 10
- Je fais la moyenne de ses notes en prenant en prenant en compte les coefficients
- Enfin, je classe les candidats en fonction de leurs moyennes

**En vrac**

Pour ma part, voici les critères qui me sont venus en tête : Linux, python, xhtml, css, javascript, ajax. Au départ, j'ai laissé cette liste telle quelle. Heureusement pour moi, le premier candidat auquel j'ai eu à faire ma fait prendre conscience d'autres critères tout aussi importants.

**La communication**

Quand je l'ai eu au téléphone, le candidat paraissait avoir beaucoup de difficulté à me comprendre. Il me faisait répéter et s'adressait à moi comme il s'adresserait au boucher du coin. Je ne pouvais pas travailler avec quelqu'un qui ne peut pas s'exprimer suffisamment bien, car chaque décision qu'il fera devra être justifiée et argumentée. Il a beau avoir d'excellentes idées ou de très bonnes raisons de faire tel ou tel choix technique ou fonctionnel, mais s'il ne sait pas communiquer une idée clairement et simplement nous ne pourrons pas valider ses choix et nous ne pourrons pas comprendre son raisonnement et on sera peut être passé à côté d'une bonne chose.

**La motivation**

Le deuxième candidat avait ceci d'intéressant qu'il s'exprimait relativement bien en français, qu'il présentait bien et qu'il était chef d'entreprise. Il est l'auteur d'une application GIS en ligne pour laquelle il a visiblement fourni beaucoup d'effort au niveau de l'interface graphique. Le candidat avait donc une expérience intéressante pour nous qui cherchions justement une application GIS en ligne. Mais voilà, il habite à 600 Km d'Alger, il ne connaît python que très peu, idem pour linux, et surtout il n'est pas du tout intéressé par le développement d'application desktop. Même si l'offre d'emploi portait exclusivement sur du développement web, il est plus intéressant pour nous de recruter quelqu'un qui puisse aussi donner un coup de main de temps en temps sur notre application desktop.

**La disponibilité**

Viens ensuite un troisième candidat qui a déjà 3 années d'expérience. L'intelligence est perceptible dans son regard, dans sa capacité d'écoute et dans ses réponses pesées, précises et sans hésitations : il est sûre de lui, il sait ce qu'il dit et il sait de quoi il parle. Malheureusement, il n'est disponible qu'au mois de Juin (c'est à dire près de 9 mois d'attente).

**La grille**

Maintenant que vous avez lister les critères importants, prenez une feuille excel, mettez les compétences en colonnes et vous réserverez les lignes pour les candidats comme ceci :

| Candidat | python | linux | disponibilité | diversité | communication | logique |
|---|---|---|---|---|---|---|
| A.B | | | | | | |
| C.D | | | | | | |
| E.F | | | | | | |
| G.H | | | | | | |
| I.J | | | | | | |
| K.L | | | | | | |

Vous remplirez cette grille pendant l'étape d'évaluation.

| Candidat | python | linux | disponibilité | diversité | communication | logique |
|---|---|---|---|---|---|---|
| A.B | 8 | 7 | 3 | 4 | 2 | 4 |
| C.D | 2 | 4 | 10 | 6 | 10 | 5 |
| E.F | 7 | 10 | 3 | 4 | 10 | 4 |
| G.H | 10 | 0 | 7 | 8 | 8 | 8 |
| I.J | 7 | 5 | 4 | 8 | 4 | 1 |
| K.L | 0 | 0 | 10 | 8 | 10 | 8 |

Si vous vous retrouvez avec des zéros dans la grille, comme c'est le cas ici, vous n'avez pas fait correctement le filtrage téléphonique dont nous allons parler juste après la recherche de candidats.

## (2) CHERCHER LES CANDIDATS

Bravo, vous avez établie votre grille d'évaluation. A présent, vous pouvez passer à l'étape suivante : la recherche de candidats.

**La méthode de recrutement standard**

Voici ce qui ce serait passé si j'étais passé par le processus de recrutement standard :

1. Je dresse une liste des compétences recherchées
2. Je rédige une annonce
3. Je la soumets aux ressources humaines
4. Ils me disent « merci, ne t'inquiète pas ! je s'occupe de tout tu s'occupe de rien. »

Ils auraient passé une annonce sur le journal et attendu plusieurs dizaines de candidatures affluer sur leur boite e-mail. Cette solution n'était pas adaptée à mon cas parce que je recherchais un profil assez rare : quelqu'un qui connaisse à la fois linux et python. Autant pour linux je pouvais en trouver quelques uns, autant pour python j'étais sûre de souffrir avant d'en trouver. Voyez-vous, le problème ce n'est pas qu'il n'y en ait pas, c'est qu'il est difficile de les atteindre, de les trouver.

**Le problème avec cette méthode**

Voici ce que je pense des recrutement par le biais d'annonce sur le journal (chiffres données à titre indicatif):

1. Vous recevez plusieurs dizaines de candidatures
2. Dont 30% sont des profils non IT (chauffeurs, électriciens, bricoleurs...)
3. Dont 2% sont des appels de personnes en détresse (aidez-moi svp, j'ai 4 enfants ; je dois acheter des médicaments ; ma mère doit faire une opération ; donnez moi un travail svp ; je suis prêt à travailler (sic) n'importe quoi...)
4. Il vous reste environ 68% de candidatures IT.

Sur ces 68% (chiffres données à titre indicatif):

1. 20% sont des profils telco
2. 60% sont des développeurs Delphi/Java/C/C++/C#/.Net et ne connaissent pas Linux
3. les 20% qui connaissent Linux sont en grande majorité des développeurs php/mysql

Sur les 20% (des 68%) qui connaissent linux, il faut trouver ceux qui connaissent python, ou qui sont capables de l'apprendre assez rapidement, par exemple ceux qui connaissent un langage qui ressemble plus ou moins à python comme Javascript ou ActionScript. (Ne rêvons pas trop, nous n'en sommes pas encore au point où des langages comme ruby et lua sont connus de notre corps professionnel dans notre pays).

En d'autre termes, vous savez à l'avance que vous allez balancer 100 C.V à la poubelle et ne garder que 5 ou 6 C.V à la fin de votre premier filtrage. Mais traiter 100 C.V c'est une perte de temps. Est-ce qu'il n'y a pas une méthode plus intéressante pour aller droit au but en évitant les C.V-spam ?

**La méthode Joël**[1]

Dans un article intitulé (LIEN VERS L'ARTICLE ICI), Joël explique très bien pourquoi il ne croit pas à cette méthode « classique ». Plutôt que d'opter pour une démarche **PUSH,** c'est à dire mettre une annonce et attendre que le poisson morde à l'hameçon, il opte pour une démarche **PULL**, c'est à dire allez directement pêcher les candidats *à la main*. Makes total sense. Quand vous passez une annonce dans le journal, à qui vous adressez-vous en premier lieu ? À des **chômeurs**, et dans cette catégorie de la population, une partie chôme parce qu'elle n'est pas assez bonne. En vous adressant aux chômeurs vous vous adressez aux personnes qui détestent leurs boulots et qui ont décidé de quitter ou aux personnes qui se sont fait virées. Je ne parle pas de personnes ayant des diplômes vs. personnes n'ayant pas de diplômes, nous avons tous faits des études supérieurs, dieu soit loué. Mais combien d'entre nous a réellement **mérité** le diplôme à la sortie ?

~~~ o ~~~

1 Joël Spolsky, entrepeneur à succès (fogs creek), était un bloggueur avec une assez bonne réputation dans la blogosphère techhnique. Il nous a donné le blog (discontinué) joelonsoftware.com ainsi que Stack Overflow.
~~~

## (longue) Digression sur le système universitaire

## et la valeur des diplômes qu'il produit

Quand j'étais en Maîtrise, un ami est venu me supplier de l'aider à traduire un texte pour lui, de l'anglais vers le français. Je n'avais vraiment pas envie de l'aider :

- ça me prenait du temps
- J'avais autre chose à faire (préparer *ma* maîtrise par exemple !)
- heck, j'étais à la fac ! J'avais envie de décompresser dans mon temps libre !

Ne voyez-vous pas que ce texte n'était rien d'autre que ce qu'il a présenté comme son mémoire de maîtrise ? Presque intégralement ? Et qu'il a eu avec mention très bien ? Oui, presque intégralement parce que, n'exagérons rien, il n'allait tout de même pas se contenter de présenter une simple traduction. **Son encadreur a fait le reste du travail lui même** en ajoutant quelques explications par ci par là, en plus des corrections orthographiques et autres fautes de styles. L'effort de l'étudiant dans tout cela ? Je n'en ai honnêtement aucune idée.

Paradoxalement, ceux qu'on croyait plutôt forts (Ils n'étaient que 3 ou 4 dans ma promo sur une cinquantaine), qu'on venaient consulter pour telle ou telle question, n'ont pas été des étudiants toujours brillants. Par exemple, 3 sur 4 d'entre eux ont eut leur licence au bout de 2 ans au lieu d'une seule. La raison ? Ils ne cherchaient pas à avoir de diplômes. Ils passaient plus de temps à s'amuser en cours qu'autre chose. Quand je dis s'amuser, je veux dire prendre du plaisir à apprendre, à approfondir un cours parce qu'ils leurs plaisaient, au détriment d'un ou de plusieurs autres s'il le faut. Poser des questions, participer, coincer le prof, le corriger au tableau, sur son propre terrain...Voilà en quoi se distinguaient ces étudiants. Ils ne prenaient pas de notes pendant que tout le monde copiait ce que le prof écrivait au tableau, ils préféraient écouter le prof, suivre son raisonnement, essayer de comprendre au lieu de parler du dernier foot en chuchotant.

Pour avoir son année de Licence, il fallait valider 12 U.E (Unité d'Enseignement) dans l'année , soit 6 par semestre en moyenne. Vous aviez le choix, vous pouviez prendre le cours qui vous plaisait, l'essentiel était d'accumuler 12 cours. Le menu était plutôt appétissant :

1. Réseaux de neurones à back-propagation;
2. algorithmes génétiques;
3. programmation parallèle;
4. synthèse d'image;
5. programmation système (sémaphores, séquenceurs, pagination etc.);
6. OpenGL;
7. géométrie computationnelle ;
8. simulations et systèmes complexes;
9. Intelligence artificielle (SAT solvers, computer-vision, programmation logique (prolog)...)
10. langages et outils spécialisés (tcl/tk, awk, sed etc.);
11. programmation de fiction (excellent cours)
12. musique et informatique
13. systèmes embarqués

14. programmation de jeux de réflexion
15. structures de données et algorithmes
16. data mining
17. informatique et Droit
18. micro-cultures sur Internet (cyber-sociologie)
19. interprète lisp (cours sur l'interprétation, il est demandé de réaliser un Interprète lisp à la fin du cours)
20. sécurité et annuaires d'accès
21. Réseau
22. + pleins d'autres cours...

Comment voulez-vous ne faire que 6 cours dans un semestre quand on vous en propose 24 ! je me souviens de ma toute première année, quand je n'étais encore qu'un ili[1], ili120, M. Bourdin, un professeur que je trouvais assez charismatique (il faisait du théâtre), nous avait dit qu'au deuxième semestre, on allait nous proposer de choisir parmi 2 langages : SmallTalk et Prolog (en plus de du C et du lisp qui étaient obligatoires). Pourquoi choisir entre ces 2 ? prenez les deux à la fois ! (4 langages en tout, en un seul semestre, en plus des autres matières).

La majorité des étudiants ont bien sûre ri parce qu'on était déjà surchargé. Les mathématiques et l'informatique demandent beaucoup, beaucoup de pratique à la maison, beaucoup d'exercices à faire et beaucoup de documentation à lire, beaucoup de patience. Ça demande trop de « beaucoups » .

Vous pouvez apprendre beaucoup d'un candidat en lui demandant de vous raconter son cursus universitaire. Certains ont choisi de prendre SmallTalk et Prolog. Voilà comment vous pouvez distinguer ceux qui sont là pour apprendre et ceux qui sont là pour un diplôme. Certains sacrifiaient volontiers un semestre entier pour s'adonner à un seul cours (quand il fallait en valider 6). Voilà comment certains étudiants ont eu leurs licences en deux ans au lieu d'une année par exemple. Je pense qu'ils ont profité plus que nous tous.

M. Degrémont, un autre professeur charismatique (oui il y en avait un certain paquet à Paris VIII ...) a été le premier à nous convaincre de l'idée du « right tool for the right job » et de l'idée que « celui qui n'a qu'un marteau dans sa boite à outil verra tous les problèmes comme des clous ». En d'autre termes, ils nous a fait prendre conscience que nous étions devenus « assez grands » pour apprendre des langages ou des environnement de programmation par nous même. C'est à ce moment là que lorsque vous entendez les commentaires de vos camarades vous les regardez plutôt comme des assistés. «Pourquoi ils ne nous font pas un cours sur le langage X, la méthode Y, le framework Z... On veut des T.P sur la techno T... ». Wait, doesn't everybody know how to use google yet ?

Ce cours m'a permis de voir concrètement qu'il y avait des outils spécialisés pour certaines tâches, et qu'on allait donc beaucoup plus vite en les utilisant qu'en codant tout à la main en C/C++/Java/You name it. Je pense que ceux qui ont joué le jeu on eut tout à y gagner.

Un ami a pris le même sujet que moi : simulation éco-touristique dans les pays de la Méditerranée. Je pense que Nicolas a fait des merveilles sur Excel avec VBA[2], il m'avait

1 ILI ou Introduction Au Langage Informatique était à la fois le premier cours qu'on prennait en classe, notre login sur les machines SUN -nous avions tous un numéro, ili120 pour moi-, et aussi notre « classe » au sens RPG du terme, ili étant le personnage le plus basique n'ayant aucun niveau d'XP.

2 Visual Basic for Applications est un langage développé par Microsoft qui s'embarque dans des applications (Embedded) et permet de les scripter.

vraiment fait comprendre tout le potentiel qui repose dans Excel, que comme quoi il n'est pas uniquement fait pour des feuilles de calcules stupides et abstraites, au contraire, toute la puissance de VBA fait d'Excel un environnement de programmation sympathique pour la programmation pilotée par les données (Data Driven Developpement).

Pour ma part j'ai découvert ce qu'étaient les SMA, les Système Multi-Agent, un paradigme extrêmement intéressant qui peut être utilisé dans tout ce qui peut être modélisé en éco-système. Ça m'a permis de conaitre l'existence d'outils tel que MadKit, 20Sim, Grass etc. J'ai passé quelque chose comme 4 mois pour la réalisation de ce projet. Ce qui m'a pris le plus de temps c'est bien entendu la rédaction du rapport et l'apprentissage du framework. Le code en lui même est celui qui a pris le moins de temps.

## Découverte d'OpenGL

OpenGL... Ça tombait mal, je venais d'apprendre python. Les deux éléments combiné ont augmenter en grande proportion le risque de prendre beaucoup de fun (et de ruiner mon semestre). Et c'est ce qui s'est effectivement passé. Python + OpenGl = Let's Rock ! Ou plutôt, let's 3D[1] ! s'il y a de vastes domaines de l'informatique que certains qualifient de « monde » (le monde linux, le monde des réseaux, le monde du web), je qualifierai personnellement la 3D, les jeux vidéos et la réalité virtuelle/augmentée *d'univers*. La aqoulou Alif-lam-mim harf, wa lakin 3D univers, jeu vidéos univers ou réalité virtuelle/augmentée univers[2]. Quelle satisfaction d'avoir accompli un loader .obj en python ! Avec support des matériaux et construction dynamique de la bounding box ! Un peu de physique avec des calculs de trajectoire (projectiles), l'éclairage, les normales, la génération de terrains... C'est vraiment un univers.

## Les lepagiens

Je pense qu'on arrive facilement à distinguer ces étudiants. Vous savez, ceux qui demandent dans un cours de maths « Donc si je comprends bien, si n >= N/2 alors la limite sera de ½ log N quand x tends vers n ? » au lieu de « Excusez-moi monsieur, c'est un plus ou un moins que vous avez écrit là-bas ? ». Je pense qu'il y a une différence fondamental entre ces deux catégories d'étudiants : certains suivent au lieu de recopier tandis que d'autres recopient au lieu de suivre.

L'intelligence se lit dans leurs regards, dans leurs questions, dans leurs réponses. Dans les cours qu'ils choisissent même. C'est peut être particulier à Paris VIII mais comme je le disais plus haut, vous aviez le droit de choisir vos cours. Et il y avait cette catégorie d'étudiants qui ne juraient que par le cours de Lepage. Dans les couloirs et dans les buvettes vous n'entendiez que « Tu as pris le cours de Lepage ? » « il faut prendre le cours de Lepage », « Le module de Lepage est très facile » etc. Lepage n'est ni une technologie, ni un concept informatique ni une technique particulière. C'est le nom d'un enseignant. Je pense que si vous demandiez aux (nombreux) étudiants qui ont pris ce cours « Quel est l'intitulé du cours ? », moins de la moitié pourront vous donner la bonne réponse. Le cours s'intitule « Ethnologie et micro-cultures sur Internet ». En deux mots : le but de ce cours était de vous apprendre à vous servir d'une imprimante. L'examen consistait simplement à imprimer n'importe quel document trouvé sur Internet et vous obteniez invariablement un 16/20. Hé oui, voilà ce que ça veut dire d'avoir un « *diplôme* ».

1 Une librairie 3D pour python que j'ai développé porte ce nom.
2 Si vous n'avez pas compris ce passage ce n'est pas bien grave. Elle n'est pas importante pour la suite de la lecture.

## Comment en est-on arrivé là ?

Voici mon propre point de vue sur la situation que nous avons actuellement :

1. Puisqu'il n'y a pas de travail, on a commencé à exiger des candidats de plus en plus diplômés, non pas parce qu'il y en avait moins mais plutôt histoire des les garder le plus longtemps possible à la fac au lieu de venir faire grossir le rang des chômeurs. La baisse du taux de chômage est l'apanage de tout politicien, des mesures pareils seront donc applaudies de tous.

2. Cette forte demande en diplômes a naturellement entraîné une forte affluence d'étudiants dans l'enseignement supérieur. Une partie d'entre eux n'étaient visiblement pas destinés à de telles études mais ont été plus ou moins forcés d'y aller par les exigences du marché de l'emploi. (DONNER ICI UN EXEMPLE CONCRET D'EXIGENCE EN DIPLÔME QUI N'EST PAS JUSTIFIÉ). Ceci a eu pour conséquence directe la diminution notoire et de la qualité de l'enseignement (nombres de places pédagogiques, organisation des séances de T.D/T.P etc.) et de la qualité de vie des étudiants (mal logés, mal nouris, mal soignés) si bien qu'un observatoire de la vie étudiante a été mis en place pour « observer » la paupérisation des étudiants en toute impuissance.

3. Enfin, cette forte affluence a eu pour conséquence que les établissements d'enseignement supérieurs, face à l'incapacité de gérer un tel flux et surtout avec un budget de plus en plus dérisoire, ce sont vus forcer de libérer des places pour accueillir les nouveaux arrivants, encore plus nombreux ! On est donc dans la légitimité de se poser des questions sur la réelle valeur de ces diplômes. Les universités n'ont pas d'autres choix car évacuer des étudiants sans leur donner de diplômes est regardé comme un taux d'échec important. Ce taux d'échec va souvent être interprété comme une mauvaise gestion et aura des conséquences sur les allocations budgétaires, sur les décisionnaires en place et sur l'autonomie de la structure. Pourquoi est-ce qu'un thésard est payé à 1600 euros ? Un enseignant-chercheur à peine à 2300 euros ? Le message est claire : « Cassez-vous le plus vite possible, ça deviens ingérable ! »

Pour toutes ces raisons, je pense que les diplômes de l'enseignement supérieur ne reflètent pas réellement les compétences des diplômés. J'ai travaillé avec quelqu'un de mon âge qui n'avait même pas son bac, et pourtant, avec ses presque dix années d'expériences (il a commencé à travaillé à l'âge de 14/15 ans) il me dépassait sur tous les niveaux en informatique. Paradoxalement, j'étais son chef parce que moi j'avais un diplôme !

~~~ o ~~~

**Retour à la méthode Jöel**

Annoncer dans un journal c'est s'adresser à des chômeurs qui ont envoyé leurs C.V à des dizaines d'entreprises différentes, quelque soit le poste proposé (ne correspondant même pas à leurs profils) et dont à peu près personne ne réponds (sauf erreur).

Au lieu de ça, Joël propose d'aller chercher les bons candidats là où ils se trouvent. Posez-vous la question de savoir :

- Sur quels sites/forums spécialisés est-ce qu'ils ont l'habitude d'aller ? → postez une annonce, passez une pub, ouvrez un thread de discussion. Chercher les sujets techniques pointus et remarquez les membres qui donnent les réponses les plus intéressantes.

- A quels salons/expositions/conférences/ateliers/workshops/séminaires/rencontres... participent-ils ? → allez les rencontrer sur place et discuter un peu avec eux. Laissez leur votre carte de visite, demandez-leur de passer au bureau. S'il y a un salon spécialisé pour votre métier, payez quelques dinars pour un petit stand, ce n'est pas cher donné pour quelqu'un à qui vous allez de toute façons verser un salaire pendant 1 an, il vaut mieux que cette année de salaire ne soit pas gaspillée à quelqu'un qui ne vous donne pas satisfaction. Si vous n'avez pas les moyens de vous payer un stand, allez-y comme simple visiteur. Il y a souvent des conférences organisées pendant ces salons, et c'est une occasion rêvée d'y rencontrer les spécialistes. Papotez un peu avec vos voisins et voyez si vous avez en face de vous un potentiel recru. Parlez aux gens autour de vous et dites que vous cherchez du monde, ils peuvent en connaître et vous recommander des personnes de leurs entourage.

- De quelles universités/écoles sortent-ils ? → postez une offre de stage à l'intérieur des campus, ça arrange les deux parties en général. Les jeunes diplômés sont souvent bien meilleurs que les chômeurs à la recherche d'un emploi et en plus c'est le moment où jamais de pêcher le poisson encore en début de carrière avant qu'il ne s'en ailles ailleurs... chez votre concurrent !

**Les banques de C.V**

Il y a aussi les banques de C.V en ligne. J'ai eu la chance de tomber sur emploialgerie.com du temps où il était 100% gratuit. C'est une véritable aubaine car en quelques heures vous avez 10 C.V à fort potentiel entre vos mains, vous n'avez plus qu'à appeler. Le grand avantage de la banque de C.V est que vous pouvez chercher des compétences bien précises et réduire ainsi le nombre de spam dans vos C.V. Dans mon cas, j'ai cherché la compétence python et j'ai eu une bonne quinzaine de résultats, le tout s'est joué sur le reste des compétences.
~~~

## (3) Premier filtrage : lecture des C.V

Bien. Vous avez votre grille et vous avez des candidats. Maintenant, il faut trier.

Si vous avez opté pour la banque de C.V, vous avez certes dépensé quelques milliers de dinars mais cette opération de filtrage a déjà été faite pour vous et vous avez gagné un temps très précieux. Si vous opté pour la méthode Joël, vous avez déjà peut être déjà trouvé votre candidat.

Si vous avez passé une annonce et que vous recevez des C.V dans votre boite aux lettres, alors retrousser vos manches et frottez bien vos yeux. Vous allez en voir de toutes les couleurs.

Cette première phase vous permettra d'éliminer environ 80% des C.V que vous recevrez. Sur 100 candidatures, 20 pourraient être pertinente.

## (4) Deuxième filtrage : l'entretien téléphonique

Sur les profils que j'ai pu voir, certains sont injoignables au téléphone. C'est handicapant. C'est handicapant car avant de convoquer le candidat pour un entretien d'embauche, j'aimerais savoir au préalable si les informations mises sur le C.V sont bonnes, légères ou si le candidat ne sait tout simplement pas de quoi il parle.

J'ai eu un candidat qui a mis « Unix » dans son C.V. Quand je l'ai appelé, et après les premiers échanges de banalités, je lui ai demandé sur quelle version d'Unix il travaillait. Après un « *heu* » hésitant Il m'a répondu version 4. Je lui ai dit «oui, mais je veux dire le Unix de quel vendeur » ? Là, il a coupé son téléphone et je n'ai pas pu le rappeler. De son côté, il n'a pas voulu donner suite à mon email.

Quelle belle opportunité également de tester ses aptitudes à l'anglais orale, comme le candidat le prétend. J'ai remarqué que nos jeunes étudiants en informatique maîtrisent mieux l'anglais que le français d'ailleurs, ce qui est une bonne chose pour. Un informaticien qui ne comprends pas l'anglais est professionnellement mort-né à sa sortie d'école. La documentation, les forums, les listes de diffusions, les salons de discussions IRC... Les sources d'information les plus importantes sont en anglais.

## (5) Troisième filtrage : tests techniques et pyscho-techniques

SCIP[1] est considéré comme un des meilleurs livres d'introduction à la programmation. Je n'ai pas lu ce livr, mais j'ai suivi les premières séances du cours éponyme donné par les auteurs, Abelson et Sussman[2]. J'ai aimé ce passge dès le début du cours, la toute première séance, les toutes premières minutes, où Sussman écrit au tableau « computer science », comme on appelait cette discipline à l'époque, puis dit quelque chose comme : computer science is not about computers (il barre le mot « computer » au tableau), and is not about science (il barre également le mot science). It's about processes to solve problems. It's not about computers in the same way that physics is not about microscopes or voltmeters, these are just the tools and instruments used by the scientists to study other subjects. It's not a science because there are no laws or theorems. It's maybe art or engineering. This « business » has more to do with magic.

C'est vraiment notre métier la résolution de problème. Nous sommes également assez bon dans la définition de processus, de workflow etc. Nous avons, dans ce contexte, pas mal en commun avec les gens qui travaillent dans l'administration, la gestion, ou l'industrie. Pour moi, un programmeur doit avoir cette capacité très abstraite de savoir résoudre des problèmes, au sens général du terme « problème ». Il y a selon moi quelques aptitudes intéressantes à rechercher chez un programmeur :

- Pattern recognition / pattern matching
- Etre capable d'analyser de différentes façons les données qu'on reçoit.
- Etre capable de suivre plusieurs chemins dans la résolution d'un problème. Il représenterait mentalement les pistes de solutions comme un graphe ou un arbre et parcourrait les branches une à une jusqu'à ce qu'il arrive à une feuille représentant une solution possible. Il est aussi capable de voir à l'avance que le chemin qu'il a pris ne le mène pas très loin et réagir en quelque sort comme l'algorithme alpha/bêta.

Voici un exemple de test que je donne aux candidats. Je ramène avec moi un jeu de carte et un jeu de domino que j'éparpille sur la table. C'est assez amusant de voir les passants nous regarder d'un air curieux, ils doivent bien se demander s'il est sérieux de jouer pendant les heures de travail.

1 Structure and Interpretation of Computer Progams.
2 Les vidéos sont disponibles en ligne sur youtube.

Je prends donc mes dominos, je les étale sur la table devant les yeux du candidats, puis j'en choisi quelques uns, et je les arrange de cette manière :

| | | | |
|---|---|---|---|
| 1 | | 6 | ? |
| 2 | 4 | 6 | ? |

Lire les dominos de gauche à droite et deviner quel domino doit se placer à droite après le double six.

Ici, le candidat doit essayer de trouver les liens entre ces dominos et compléter la suite logique. Ce que je cherche ce sont les candidats qui arrivent à raisonner devant un problème donné. J'ai été deçu de la plus part d'entre eu car ils ne raisonnent pas. Vous savez qu'à l'université où j'ai été nous avions tout un cours dont le but était de nous apprendre à raisonner. Le professeur nous donnait les outils nécessaires à la réflexion tels que l'analyse, la déduction, l'induction, la récurrence etc. Pour s'exercer, nous avons pris des problèmes mathématiques classiques et nous avons en quelque sorte redécouvert des solutions à des problèmes connus depuis longtemps, par exemple la re-découverte par nos propres moyens du nombre imaginaire $i$ par résolution d'une certaine forme d'équation polynomiale du troisième degré[1].

Voici le type de raisonnement que j'attends de mes candidats : est-ce que les nombres se suivent en ajoutant 1 ? (1,2,3,4...) ; Est-ce qu'il diminuent de deux ? Si j'arrive à 0, est-ce que je repart de 6 ? est-ce que je dois chercher une logique entre le chiffre inscrit en haut du domino et celui du bas ? Dois-je regarder verticalement ou horizontalement ? Si j'essaye l'horizontal, est-ce que c'est une multiplication que je dois faire ? Etc.

Les candidats qui réussissent ce test me prouvent qu'ils savent essayer plusieurs solutions au lieu de se contenter d'un « je ne sais pas » au bout d'une ou deux tentatives. Ils me prouvent qu'ils savent représenter un problème dans leurs têtes, qu'ils savent raisonner. Ces aptitudes sont les mêmes que celles qu'on utilise dans la phase la plus longue du cycle de développement logiciel : le débogage. On doit mettre en correspondance les données éparpillées sur les fichiers de logs, la sortie d'erreur, la sortie standard et celles qu'on manipule en entrée, et tout comme ce test de dominos, on doit trouver des liens entre les valeurs de ces données et déduire la logique qui nous permettra de localiser avec plus ou moins de précision dans quel « espace » se trouve le bug. On m'a judicieusement fait remarquer que dans notre métier de programmeur, nous ne faisons de la programmation que pendant les 5 à 10 premières minutes d'un projet. Dès les premières minutes, on est déjà en train de tester notre code et on s'aperçoit des nos premières erreurs. Au bout de seulement quelques minutes, on est déjà en train de débugger. Je fais ça tous les jours mais je n'ai jamais pris conscience de la précocité du débogage dans notre travail, ça intervient vraiment dès le début du projet. Bien sûre, au début, quand on a écrit qu'une centaine de lignes et seulement une fonctionnalité, on sait en une fraction de seconde où se trouve l'erreur s'il y en a. Mais au bout de 2 mois de travail et après quelques milliers de lignes de code, c'est déjà beaucoup moins évident de trouver le problème.

J'ai eu un candidat qui maîtrisait très bien les technologies. Ses connaissances en XML étaient vraiment appréciables. Finaliste Imagine Cup 2010, il a travaillé avec son équipe sur un projet fort intéressant dans le domaine de l'IHM[1] : ils ont implémenté des fonctionnalités au système d'exploitation windows permettant par exemple d'ouvrir un dossier en clignant de l'œil, de naviguer en faisant des gestes simples avec les mains (à droite, à gauche) etc. Le candidat était disponible de suite, mais il n'a réussi aucun des test logiques que je lui ai soumis. Il n'a pas abandonné, ce qui est

1 $x^3 + x^2 + x + 1 = 0$

1 Interface Homme-Machine

une bonne chose, malgré les contraintes de temps qu'il avait (il était passé entre 12h00 et 14h00 pendant sa pause déjeuner, étant déjà employé pas très loin), mais même en lui donnant beaucoup d'indices, il n'a pas réussi à solutionner les puzzles, peut être par manque de concentration.

J'ai eu un autre candidat qui avait de très bonnes facultés intellectuelles. Il a solutionné tous les puzzles que je lui ai donné très rapidement et avec une assurance déconcertante. Je lui ai demandé s'il était bon en maths, il m'a répondu que ce n'était par pour se vanter mais il avait des 17 en maths et d'ajouter « bessa7 makan walou, mayakhal3ekch »[3]. Seulement voilà, il connaissait python mais sans l'avoir vraiment appris. Je ne lui en veut pas, je lui tire même chapeau d'avoir réussi à programmer dans un nouveau langage sans avoir accès à Internet, car ceci implique que s'il est bloqué sur une ligne de code, il ne peut absolument rien faire pour rechercher de l'aide. Le problème quand on programme dans un nouveau langage sans vraiment l'apprendre : on « pense » dans le langage avec lequel on a travaillé précédemment. Pour beaucoup de gens, ça sera Java.

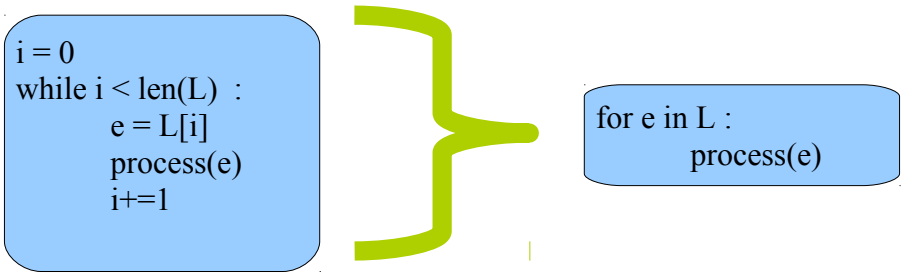
~~~ O ~~~

### (courte) Digression sur les langages de programmation

Apprendre à programmer en python c'est apprendre à penser en python. C'est la même analogie qu'avec les langues : apprendre l'anglais c'est apprendre à s'exprimer en anglais, à utiliser les idiomes anglais. Par exemple, pour dire : « servez-vous, je vous prie » en anglais vous ne dites pas : « please serve yourself » mais « please help yourself ». De même, l'anglais vous offre de nouveaux outils pour vous exprimer plus directement que vous ne le feriez en français. Par exemple, vous pouvez utiliser « it sucks » à chaque fois que vous jugez quelque chose comme médiocre ou qui n'a rien d'intéressant. « this website sucks », « this article really sucks ». Le mot le plus proche en français serait peut être «c'est naze » ou « ça craint ».

J'ai demandé à un candidat de m'écrire une code python qui donne les 100 premiers mlutiples de 3. Le candidat a bien écrit en python, mais ça ressemblait à du code java. : while, i+=1 , if, else... Le code python, dans le style python est : range(3,300,3).

**PENSER EN PYTHON**




3 Qu'on pourrait traduir par « it's no big deal, really »
~~~

Pour apprendre un nouveau langage, il ne suffit pas de connaître la syntaxe et les fonctions à appeler, il faut aussi « lire » du code écrit par des experts dans ce langage, tout comme nous faisons beaucoup de lecture pour apprendre une nouvelle langue. Il e suffit pas de traduire littéralement une expression du français pur obtenir l'équivalent en anglais. Ça ne marche tout simplement pas comme ça. Il faut également « parler » ce langage, avec des personnes parlant nativement cette langue si vous avez la chance d'en connaître, sinon, comme pour la plus part d'entre nous, dans les forums, les salons IRC dédiés et les listes de diffusion, parce que même si vous lisez beaucoup, ça ne suffit pas pour apprendre ***à s'exprimer***. Quand vous parlez ce langage, les experts vous corrigeront et vous aideront à mieux vous ***exprimez***.

~~~ o ~~~
~~~

## (6) EVALUTATION DES CANDIDATS

Vous y êtes presque ! Vous avez passé des entretiens à tous vos candidats et il ne vous reste plus qu' à les évaluer maintenant.

Mes critères étaient les suivants :

linux ; python ; disponibilité ; communication ; diversité ; logique ; technique ;

| Candidat | python | linux | disponibilité | diversité | communication | logique | technique |
|---|---|---|---|---|---|---|---|
| A.B | | | | | | | |
| C.D | | | | | | | |
| E.F | | | | | | | |
| G.H | | | | | | | |
| I.J | | | | | | | |
| K.L | | | | | | | |

Y avait-t-il un critère plus important qu'un autre ? Sur lequel devrais-je porter mon attention ? Comment pondérer ces critères ?

Voici la méthode que j'ai suivi pour établir mon ordre de priorité :

- j'ai écrit mes critères dans un traitement de texte, open office dans mon cas, dans une liste numérotée.
- Ensuite j'ai simplement fais un tri à bulle.

**Le tri à bulle**

Ce tri est appelé ***tri à bulle*** parce que les critères les plus importants sont ramenés progressivement « à la surface », en haut de la liste, comme une bulle d'air dans l'eau. Voici comment ça marche. Supposons que votre première liste ressemble à ceci :

1. communication
2. diversité
3. python
4. logique
5. linux
6. technique
7. disponibilité

Nous allons comparer le premier élément de la liste avec le suivant. S'il est plus important, il reste en haut, sinon, si c'est celui du bas qui est plus important, il prends la place de celui qui est en haut et celui qui est en haut descends.

Est-ce que la communication est plus importante que la diversité ? Parfois les choix ne sont pas faciles à faire quand les critères paraissent d'égale importance, mais voici une astuce que j'emploie souvent et qui marche très bien : si j'ai quelqu'un qui s'exprime très bien, mais qui ne s'est pas assez diversifié, est-ce que je le prendrais (en faisant abstraction des autres critères) ? . Hum... je pense que oui parce que bien s'exprimer est plus important, la diversité est un plus. Maintenant je pose la question inverse : si j'ai un candidat très diversifié mais qui s'exprime très mal, est-ce que je le prendrais ? Ah ! Certainement pas ! Comment pourrais-je profiter de cette diversité s'il s'exprime mal à propos ? Hum... donc, je conclus que la communication passe devant la diversité, et c'est pour cela qu'elle reste pour l'instant en tête de liste. Ma technique consiste donc à me mettre en situation et à imaginer le candidat. S'il avait la qualité A mais pas la B, est-ce que je le prendrais ? Si je n'arrive pas à répondre oui ou non je pose la question inverse, s'il avait la qualité B mais pas la A, est-ce que je le prendrais ? et dans, la majorité des cas si la première question n'a pas réussi à trancher, la deuxième réussi presque toujours. Si vous n'arrivez vraiment pas à départager, vous avez le droit de considérer les deux qualités/compétences comme égales, vous leur donnerez alors le même coefficient. Ceci va dépendre de vous, du poste pour lequel vous cherchez un candidat, de votre entreprise et de votre domaine d'activité.

Si par contre vous mettez la même importance, si vous mettez au même rang deux critères parce qu'ils vous paraissent tous ls deux de moindre importance, posez-vous la question de savoir si ce sont des critères pertinents en premier lieu ? Peut être devriez-vous simplement les écarter.

Je continue donc, ma liste est restée telle quelle :

1. communication
2. diversité
3. python
4. logique
5. linux
6. technique
7. disponibilité

J'ai comparé le 1 avec le 2. Maintenant je compare le 2 avec le 3. Est-ce que la diversification est plus importante que python ? Prenons un candidat qui ne connaît pas python du tout, mais qui a fait autre chose dans sa carrière. Etant donné que le poste est pour un développeur python en premier lieu, il est donc fondamental que cette personne connaisse le langage. Adjugé ! Le 3 est plus important que le 2, il doit donc remonter. J'utilise pour cela les boutons spécifiques à OpenOffice, je ne les ai pas trouvé dans MS Office mais peut être que j'ai mal cherché.

1. communication
2. python
3. diversité
4. logique
5. linux
6. technique
7. disponibilité

L'algorithme du tri à bulle me dit que lorsqu'un élément monte dans la liste, il doit être comparé à l'élément au dessus de lui et continuer son ascension s'il est plus important. Puisque python est monté, je dois le comparé à la communication. Avec la même méthode, je considère que python est plus important et je le monte encore plus haut.

1. python
2. communication
3. diversité
4. logique
5. linux
6. technique
7. disponibilité

C'est bon, ma première bulle est remontée à la surface ! Mais le tri n'est pas terminé puisque je n'ai pas encore fais tous les éléments de la liste, je dois atteindre le tout dernier pour que le l'algorithme s'arrête.

Je recommence donc les étapes en repartant du haut vers le bas. Pas la peine de comparer python avec la communication puisque je viens de le faire, donc je compare 2 et 3, mais comme c'est déjà fait je garde tel quel. Ensuite je compare 3 et 4, diversité et logique etc. S'il y a un élément qui monte, il doit continuer son ascension tant qu'il est considéré plus important que celui qui le précède, comme nous l'avions dit, telle une bulle d'air qui remonterait jusqu'à la surface. Cette bulle d'air s'arrête si jamais il y a un élément plus important qu'elle, auquel cas elle dois s'arrêter juste en dessous. C'est comme ça par exemple que linux est remonté juste en dessous de python.

**Pondération des critères**

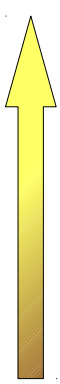
Après le tri à bulle, vous vous retrouvez avec une liste où les critères sont triées du plus important au moins important, par exemple :

1. technique
2. python

3. linux
4. disponibilité
5. communication
6. logique
7. diversité

Ma méthode consiste à reporter cette liste sur votre grille d'évaluation, de mettre un poids de un pour le dernier (le 7), et d'incrémenter de 1 en remontant la liste. Si deux critères sont considérés égaux, donnez-leur le même poids. Libre à vous d'incrémenter de plus de 1 si vous pensez que le critère mérite plus d'importance. Il aura en conséquence un plus grand impacte sur le moyenne du candidat.

| | |
|---|---|
| technique | 6 |
| python | 5 |
| linux | 4 |
| disponibilité | 3 |
| communication | 2 |
| logique | 1 |
| diversité | 1 |



**Calcul des moyennes**

| Candidat | python | linux | disponibilité | diversité | communication | logique | technique | Moyenne |
|---|---|---|---|---|---|---|---|---|
| A.B | 7 | 8 | 4 | 9 | 3 | 4 | 1 | 4.91304348 |
| C.D | 8 | 3 | 1 | 2 | 3 | 4 | 4 | 4.04347826 |
| E.F | 5 | 4 | 5 | 7 | 1 | 6 | 4 | 4.43478261 |
| G.H | 2 | 4 | 3 | 7 | 4 | 7 | 9 | 5.13043478 |
| I.J | 5 | 1 | 6 | 6 | 4 | 1 | 5 | 4.26086957 |
| K.L | 7 | 7 | 7 | 9 | 5 | 3 | 6 | 6.56521739 |
| **Coéfficients** | | | | | | | | **Total** |
| | 5 | 4 | 3 | 2 | 2 | 1 | 6 | 23 |

Placez les notes de vos candidats dans chaque case et calculer une moyenne en faisant la somme de toutes les notes multipliées par leurs coefficients respectifs, que vous diviserez par la somme des coefficients.

Exemple pour le premier candidat : 7*5 + 8*4 + 4*3 + 9*2 + 3*2 +4*1 +1*6 / 23 = 4,91

**Une véritable aide à la décision**

Cette technique m'a vraiment aidé dans le choix de mon candidat parce que malheureusement aucun d'eux n'était parfait (fallait-il s'y attendre ?). Soit il était techniquement très bon, mais s'exprimait très mal, soit il était très intelligent mais avec une attitude, soit il s'exprimait très bien mais manquait d'esprit logique... En tant que recruteur on essaye de trouver un candidat qui « ait la moyenne » dans tous les domaines. Avec cette méthode, j'ai pu trier mes candidats par moyenne obtenue, il ne me restait plus qu'à prendre le premier ou de « passer au suivant » quand le premier n'était soit plus disponible (a préféré partir ailleurs), ou avait changé d'avis, ou posait problème à la direction (souvent un problème de salaire).

| Candidat | python | linux | disponibilité | diversité | communication | logique | technique | Moyenne |
|---|---|---|---|---|---|---|---|---|
| K.L | 7 | 7 | 7 | 9 | 5 | 3 | 6 | 6.57 |
| G.H | 2 | 4 | 3 | 7 | 4 | 7 | 9 | 5.13 |
| A.B | 7 | 8 | 4 | 9 | 3 | 4 | 1 | 4.91 |
| E.F | 5 | 4 | 5 | 7 | 1 | 6 | 4 | 4.43 |
| I.J | 5 | 1 | 6 | 6 | 4 | 1 | 5 | 4.26 |
| C.D | 8 | 3 | 1 | 2 | 3 | 4 | 4 | 4.04 |

Ça me permettait aussi de faire un choix plus ou moins fondé et contraire à mes propres préférences : le candidat qui avait la meilleur note n'était pas celui que je préférais, et pourtant, il était statistiquement meilleur que les autres ! Si je voulais être objectif, il fallait que je le prenne lui, et pas mon favori.

Libreoffice est assez intelligent pour trier tout le tableau en fonction de la dernière colonne. Il suffit de sélectionner toutes les cellules de votre tableau et de cliquer sur l'icône Z-A pour avoir le classement de vos candidats.

**Pour finir**

Comme tout bonne pièce de théâtre, le doute doit conclure en beauté la tragédie jouée sur les planches. Dans mon cas, la direction général a tout simplement refusé le candidat sélectionné parce qu'il avait moins de trente ans. Tous les candidats que j'ai vu avaient moins de trente ans. Le comble c'est que cette même direction général m'a proposé de prendre quelqu'un sans expérience, « ma 3lich »[1], parce qu'elle nous voulait tout simplement pas payer un salaire d'informaticien mais un salaire d'agent de saisie en gros. Donc si je comprends bien, vous voulez recruter quelqu'un qui a 30 ans et qui n'a jamais travaillé. Heu...Je serai à votre place je me demanderais bien pourquoi cette personne n'a jamais travaillé, et douterais légitimement de ses capacités professionnelles.

1 Qu'on peut traduire par « ce n'est pas bien grave ».

Finalement, quelques semaines plus tard, je croise le directeur général dans les couloires et l'intercepte  pour lui faire part de cette restriction paradoxale. « Puisque nous prenons une agence web pour faire le site, il n'y a pas lieu de recruter une personne en plus ».

Jeu, set et match !

# (7) Troisième entretien avec le DRH

Mon histoire s'est terminée ici, mais j'espère que pour vous elle ira jusqu'au bout.

Quand vous aurez lister vos candidats par ordre de préférence, le DRH pourra les convoquer et leur faire passer son entretien d'embauche. Votre job en tant que technicien était d'évaluer techniquement les candidats. Le job du DRH est d'évaluer son attitude, sa psychologie, sa motivation, ses aspirations et sa disposition ou pas pour le poste. Mais là encore il y a des petites astuces que pas tous les DRH connaît forcément.

Dans cette entreprise où j'étais, on demandait à tous les candidats qui venaient passer un entretien de remplir une fiche de renseignement. C'était peut être une manière pour les ressources humaines de prouver qu'ils travaillaient mais là n'est pas la question. C'était une bonne chose pour moi car l'écriture d'une personne peut révéler certains traits de son caractère

**La psychognomie**

La psychognomie est une pseudo-science décrite par un certain Dr. Joseph De Galles. Je pense que c'est la seule personne à avoir développer cette technique. En réalité elle n'est que l'addition de la physionomie, de la frénologie et de la graphologie. La physionomie et la frénologie donnent les traits de caractères innées d'une personne, tandisque la graphologie donne les traits de caractères acquis. Les spécialistes en physionomie vous diront par exemple qu'une personne au visage ovale a des traîts de caractères nobles, tandis qu'une tête carrée est plutôt faite pour une carrière militiare. Un visage mince présente une personne asténique, physiquement faible, un visge rond et large est plutôt le signe d'une personne douée pour les travaux physiques (sportifs). Ceux de la frénologie vous diront qu'une personne présentant une protubérance visible au devant de la tête possède de bonnes facultés intellectuelles, tandis que ceux qui ont un crâne plat en manquent. Ceux ayant une protubérance sur les tempes ont tendance à idéaliser (artistes) ceux ayant une bosse juste au dessus des oreilles ont ce qu'ils appellent la bosse de la destructivité (des gens à fuir !). Les graphologistes vous diront qu'un « o » et un « a » hérmetiquement fermés proviennent d'une personne plutôt secrète, tandisque qu'une barre de « t » très haute dans la graphie est le signe d'une personne très

ambitieuse et un point de i lancé dans les airs est le signe d'une personne rêveuse, romantique. Des traits angulés et édentés sont la carractérristique d'une personne vigoureuse, énergique. Une écriture ondulée, en courbes, bouclée, montre un caractère doux.

La psychognomie fait une somme de tout ça. Parfois la graphologie corrige la phrénologie, c'est à dire que la personne à travailler son « défaut » inné et l'a corrigé au fil du temps, et parfois elle exagère ou confirme encore plus le trait, par exemple une personne présentant la protubérance de l'idéalité avec une graphologie qui vient confirmer son esprit rêveur, romantique.

**Body language**

Une personne qui croise les bras en s'adossant dans son siège montre sa désaprobation. Une personne mettant ses mains à plats sur la table, dirigés vers vous, motre une personne collaborative. Si les mains sont trop avancées vers vous elle cherche à vous conquérir, elle domine votre territoire. Un index sur la tête ou au dessus de la lèvre montre une personne intellectuelle.

Le pouce et le doigt autour du menton montre une personne qui se pose des questions, porté vers la réflexion. Une personne s'adossant au fauteuil avec le coude qui dépasse et une jambe sur l'autre montre des signes de prétention.

Il y a donc tellement de choses que vous pouvez vérifier chez votre candidat rien qu'en examinant sa physionomie, sa tête, son écriture et ses gestes. Bien entendu, il est toujours indispensable de vérifier ces informations par des moyens détournés, sans que votre interlocuteur ne soupçonne que vous êtes en train de le sonder. Seuls les bons spécialistes connaissent ces techniques.

## (8) Signature du contrat

Sur les C.V que vous allez consulter, vous allez trouver des personnes qui sont déjà en poste ou encore des personnes qui sont parties à l'étranger. Il est toujours bon de garder les C.V bien rangés même après avoir fait votre choix, je dirais même plus, *après* que vous ayez officiellement recruté et que le candidat est devenu votre collègue, et ceci pour des raisons qui peuvent ne pas paraître évidente au premier coup d'œil, comme ça a été le cas pour ma direction : votre nouveau collègue a le droit à une période d'essai pendant laquelle tout est possible. Soit vous vous rendez compte que vous n'avez pas assez bien évalué votre candidat, soit le candidat trouve une offre plus intéressante et décide de vous quitter, soit il peut changer de lieu d'habitation et se retrouver très loin de son lieu de travail, dans l'impossibilité de venir travailler (changement de ville, de région, de pays etc.), soit il peut décider de continuer ses études... Tous les scénari sont possibles et rien n'est gagné d'avance. C'est pourquoi il est imprudent d'arrêter les entretiens d'embauche au premier candidat qui vous a convaincu. Même s'il a tout ce qu'il faut, même s'il est parfait, prenez la soin de voir les autres.

# FIN.

Crédit photo :
Office space, de mike judge (1999)
Idiocracy, de mike judge (2005)